

Neural Network-Based Face Detection with Partial Face Pattern

Sinan Naji¹, Roziati Zainuddin¹, Hamid A. Jallb¹, Masoud Abou Zaid², and Amar Eldouber²

¹Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

²Faculty of Education, Al-fatah University, Tripoli, Libya

Abstract — *In this paper, we present a neural network-based method to detect frontal faces in grayscale images under unconstrained scene conditions such as the presence of complex background and uncontrolled illumination. The system is composed of two stages: threshold-based segmentation and neural network-based classifier. Image segmentation using thresholding is used to reduce the search space. Artificial neural network classifier would then be applied only to regions of the image which are marked as candidate face regions. The ANN classification phase crops small windows of an image, and decides whether each window contains a face. Partial face template is used instead of the whole face to make training process easier. To minimize the probability of misrecognition, texture descriptors such as mean, standard deviation, smoothness and X-Y-Relieves are measured and entered besides the image as input data to form solid feature vector. The ANN training phase is designed to be general with minimum customization and to output the presence or absence of a face (i.e. face or non-face). In this work, partial face template is used instead of the whole face. Aligning faces is done using only one point that is “face center”.*

Keywords: Thresholding, image segmentation, artificial neural network, texture analysis, and face detection.

1. INTRODUCTION

Human face detection is becoming a very important part of many vision-based systems such as face recognition, face tracking, video conferencing, etc. The first task of such systems is to locate the face (or faces) in the given image. It is not an easy task since human face is non-rigid object. Differences such as various facial expressions, presence or absence of structural components (e.g. beard, mustaches and glasses), illumination variations, number of faces, pose, size, location, rotation, complex background, etc. complicate face detection.

Numerous techniques have been developed to detect faces in images such as principle component analysis (PCA) [17][20], neural networks [2][13], Bayesian classifier [16], template matching [6], support vector machines [12], skin color [4][23][24], shape information [21], hidden Markov model [11], fuzzy rule base system [10], etc. Each of these methods has its own advantages and disadvantages. Extensive survey on detecting faces in images is presented in [22][5]. Yang et al. [22] classified known methods for face detection into four categories: knowledge-based

methods, template matching methods, feature invariant methods, and appearance-based methods where some methods overlap category boundaries.

Artificial Neural Networks (ANN) had been applied successfully in many pattern recognition problems. Since face detection can be treated as a two class pattern recognition problem (i.e. face and non-face), various neural network architectures have been proposed [2][7][9][13][14]. The face templates are learned from a set of face training images. These learned templates are then used for detection.

The main goal of this work, inspired of Rowley's work [13], is to increase the reliability of neural network-based face detector, speed up the system, and improve the accuracy. The system is composed of two stages: threshold-based segmentation and neural network-based classifier using partial face pattern. Image segmentation using thresholding is used to reduce the search space. Artificial neural network classifier would then be applied only to regions of the image which are marked as candidate face center regions. Image segmentation is presented in Section 2. Section 3 presents the detailed description of ANN training phase. Classification phase is presented in Section 4. Eliminating the overlapped detection is presented in section 5. The positively encouraging results obtained and conclusions are given in Section 6 and 7 respectively.

2. Image Segmentation

The main drawback of neural networks-based classifiers or other appearance-based methods is the huge search space at run time. These systems are very computationally demanding and therefore are not very interesting in applications where real-time or near real-time performance is required. According to Rowley's work [13], “To detect faces anywhere in the input image, the filter (i.e. window) is applied at every location in the image. To detect faces larger than the window size, the input image is repeatedly reduced in size (by subsampling), and the filter is applied at each size”. The size of the filter is 20 x 20 pixels. Nowadays, high-resolution digital cameras are available, leading to having enormous number of windows cropped from every location in the image and at several scales.

In this work, simple heuristic such as the fact that facial features differ from the rest of the face because of their low brightness, is used to speed up the system. This heuristic can decrease the search space because facial

features and other low brightness objects will be thrown out.

We propose to use threshold-based image segmentation as a preceding stage to generate new search space. “Thresholding is a fundamental approach in image segmentation that enjoys a significant degree of popularity, especially in applications where speed is an important factor” [15]. The new search space g is a binary image of same size as source image and obtained as follows:

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases}$$

Where T is calculated locally. The ANN classifier would be then applied only to locations which are marks as 1’s instead of the whole image which would speed up the algorithm as well as eliminating false detections. It turned out, by considering 205 images, that the search space is reduced by 30% to 52% following this method.

3. Artificial Neural Network Classifier with Partial face Pattern

Various neural network architectures have been proposed [2][7][9][13][14]. According to our knowledge, all these methods used whole face pattern. One of the challenging problems in the training phase is the high variability in face appearance. It is clear that reducing the variability will be helpful. Standard deviation (ς) is used to figure out the most variable facial features. It is calculated as in [20]:

Step 1: Obtain face images I_1, I_2, \dots, I_m (training faces)
the face images must be *centered* and of the same size.

Step 2: Represent every image I_i as a vector Γ_i

Step 3: Compute the average face vector Ψ :

$$\Psi = \frac{1}{m} \sum_{i=1}^m (\Gamma_i)$$

Step 4: Compute the standard deviation vector (ς):

$$\varsigma = \sqrt{\frac{\sum_{i=1}^m (\Psi - \Gamma_i)^2}{m - 1}}$$

The standard deviation vector (ς) shows that the lower part of the face has higher variations (i.e. the presence or absence of structural components such as beard and mustache combined with high degree of deformability of mouth make training harder). In this work, partial face pattern is used instead of whole face. Our face pattern contains only eyebrows, eyes, cheeks, down to nose tip. The size of face pattern is (15×23) pixels as shown in figure 1. Excluding the lower part of the face makes the training process easier as well as eliminating many false negatives.

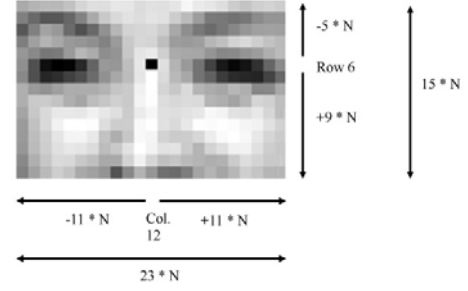


Figure 1. Sample of face template.

The size of face pattern is (15×23) pixels with face center at location (6,12).

Automatic face detection using ANN or any appearance-based face detection methods such as PCA (Eigenface), SVM, HMM, etc. have to deal with *aligning problem*; i.e. aligning all images so that the approximate positions of the facial features are the same. Many researchers carry out this task manually. Rowley et al [13][14] handled this problem by labeling many points in each training face (i.e. eyes, tip of nose, and corners of the mouth). Then, these points were used to normalize each training face to the same scale, orientation, and position. From our point of view, this normalization causes distortion to some faces since faces differ from one to another. In this work, aligning faces is done using only one point. We refer to this point as “face center”. It is defined as the intersection of two lines: the first one passes horizontally through both eyes and the other vertically upwards from the tip of the nose.

The position of each facial feature relative to that point is considered in preparing our training faces. The location of the face center would be at pixel (6,12). Figure 1 shows the geometrical information relating facial features relative to the face center. By using the distance ratios, the method becomes relatively insensitive to the size of the face with respect to distance from camera. As we will see later in training phase (section 3.2), to crop faces of different sizes, the initial value of N is 1 that incremented repeatedly by some fixed value.

3.1 Texture Analysis

“When solving a pattern recognition problem, the ultimate objective is to design a system which will classify unknown patterns with the lowest possible probability of misrecognition” [25]. The main complexity in training neural networks is exacerbated by the fact that we are dealing with huge chunks of data. Consider our (15×23) pixels face pattern; we have 256^{345} possible combination of gray values. It is extremely high dimensional space. Even with a lot of training data, there is a probability of misclassification. The main reason lies in the fact that each pixel’s intensity, x_i , represents the i^{th} descriptor in the feature vector without any consideration to the spatial relationships between them.

In this work, we propose to create a solid feature vector, in which not only the intensities of pixels are considered but also the content of neighboring pixels (i.e. spatial relationships between them). It is assumed that human faces have a distinct texture and the same type of facial features will have the same brightness. Therefore, a set of texture descriptors are measured from each face image and entered along with the image as input data to the feature vector. An important approach for describing a region is to quantify its content using statistical properties. Mean (m), standard deviation (σ) and smoothness (r), adapted from [3], are measured for predefined regions (i.e. eyes, cheeks, nose forehead, and nose tip). Smoothness measures the relative smoothness of the intensity in a region. These descriptors are defined as follows :

$$m = \sum_{i=0}^{L-1} z_i p(z_i)$$

$$\sigma = \sqrt{\sum_{i=0}^{L-1} (z_i - m)^2 p(z_i)}$$

$$r = 1 - 1/(1 + \sigma^2)$$

where $p(z)$ is the histogram of the intensity levels, and L is the number of possible intensity levels [3].

Then X-Y-Relieves, adapted from [1] [8], are determined by processing the horizontal and vertical profiles of the face pattern. Y-relief is obtained by summing all pixel intensities in each row. Similarly, X-Relief is obtained by summing all pixel intensities in each column:

$$HI(x) = \sum_{y=1}^n f(x, y), \quad VI(y) = \sum_{x=1}^m f(x, y)$$

It is easy to locate facial features by detecting the local maximum (or minimum) and first abrupt transition. As shown in figure 2, it is assumed that each facial feature generates a maximum in Y-Relief and has specific X-Relief characteristics.

Throughout the texture descriptors extraction stage, twelve descriptors are measured. These are:

- M_1 : smoothness of the left cheek region.
- M_2 : smoothness the right cheek region.
- M_3 : mean darkness ratio of left eye darkness to left cheek.
- M_4 : mean darkness ratio of right eye darkness to right cheek.
- M_5 : mean darkness ratio of left cheek darkness to right cheek.
- M_6 : mean darkness ratio of left eye darkness to nose forehead.
- M_7 : mean darkness ratio of right eye darkness to nose forehead.

- M_8 : mean darkness ratio of left cheek darkness to nose tip.
- M_9 : mean darkness ratio of right cheek darkness to nose tip.
- M_{10} : local minimum, Y-Relief for eyes.
- M_{11} : local minimum, Y-Relief for nose tip.
- M_{12} : local maximum, X-Relief for nose forehead.

Texture descriptors, M_i , are entered beside pixel intensities, x_i , as input data to form the feature vector. By using the ratios of darkness between selected facial features, the method becomes relatively insensitive to the illumination.

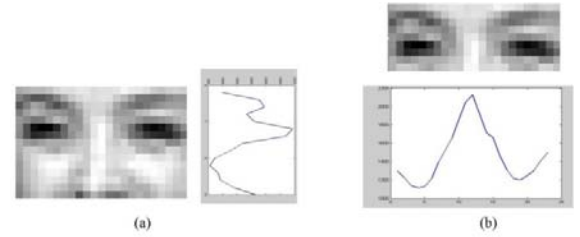


Figure 2. X-Y-Relieves. (a) Y-relief. (b) X-relief

3.2 ANN Training Phase

Generally, training phase needs a lot of face and non-face images. A classifier is trained using standard multilayer backpropagation neural network from a database of 15,200 image patterns. There are 3,200 positive examples of face patterns and the rest are non-face patterns. The training phase involves two stages: extracting texture descriptors, and training the classifier to learn the feature vector. Our face images are prepared as follows:

- 1) Faces are cropped manually from images. Usually, source images contained faces of various sizes, orientations, positions, and intensities. All training faces are frontal faces rotated up to $\pm 10^\circ$.
- 2) "Face center" is labeled manually for each face.
- 3) Many sub-images are cropped automatically using a small program called "cropper". The function of the cropper is to crop a sub-image window using a predefined information relative to face center. To crop faces with different size, the cropper increases the window size repeatedly by ratio of 1.1 and crop a sub-image window at that size for each step; leading to a pyramid of faces. This will guarantee that the location of face center would appear at same predetermined location in all faces. Then, move the face center one pixel in all directions and repeat step 3.
- 4) Sub-images that meet certain level of similarity are selected manually for the next step.

- 5) All sub-images are resized to standard size of (15×23) pixels.
- 6) In order to reduce variability due to lighting conditions and camera gains; histogram equalization is applied to improve contrast.
- 7) Texture descriptors are measured and entered beside each training image as input data.

Generally, the problem is how to describe and characterize “non-face” images. We can say that any sub-image not containing a face can be characterized as a non-face image. This makes the space of non-face images very large compared to the face images.



Figure 3. Samples of face and non-face images used for training

Instead of collecting the non-face images by hand, non-face images are collected as the following:

- 1) A set of source images are selected at random.
- 2) Demolish all faces to obtain images of scenery which contains no faces.
- 3) Run the system to crop 10,000 sub-images at random window sizes and locations.
- 4) Resize sub-images to 15×23 pixels.
- 5) Apply the preprocessing step to improve contrast and normalize intensities.
- 6) Images that resemble faces are discarded.
- 7) Texture descriptors are measured and entered beside each image as input data.

During training and validation stages, false detections are added to the training database as new non-face examples. About 2000 non-face examples are added in this way. Figure 3 shows samples of face and non-face images used for training.

The collected training images are divided into three subsets: Training, validation, and testing. The training set is used first to learn and train the ANN, the validation set is used to further refine the ANN architecture. The test set is used to measure the performance of the ANN.

There are 357 neurons ($345 \text{ pixels} + 12 \text{ texture descriptors}$) at the input layer, each one representing one feature of our feature vector. Ten neurons in the hidden layer 1; five neurons in the hidden layer 2 and one neuron in the output layer which generates an output ranging from -1 to +1, signifying the presence or absence of a face, respectively. The network's weights in hidden layers are initialized with random values. Then, face and non-face images are repeatedly presented as input with the corresponding

desired targets. The output is compared with the desired target, followed by error measurement and weights adjustment until the correct output for every input is reached. The hidden layers neurons are estimated using activation functions that feature the log-sigmoid transfer function, whereas, the output layer neuron is estimated using the activation function that features the linear transfer function.

4. ANN Classification Phase

The ANN classification phase consists of four steps: the cropper, histogram equalizer, texture descriptors extractor and ANN classifier. The first component, the cropper, receives two images of the same size: the source image (i.e. gray scale) and a binary image in which the corresponding new search space is defined. In the search space image, each pixel with value of 1 is considered a face center candidate. Pixels of value 0 are ignored. The function of the cropper is to crop a sub-image window from the source image, at every corresponding location in search space, and passes it to the next step. To detect faces with different size, the cropper increases the window size repeatedly in the same manner as in training phase; leading to a pyramid of sub-images, see figure 4. To detect faces anywhere in the input image, the cropper moves to every location in the search space and starts cropping. Then, each cropped sub-image is resized to 15×23 pixels. Histogram equalizer improves contrast. Texture descriptors are measured and then entered as input data beside each sub-image to the ANN classifier, which decides whether it contains a face or not. It generates an output ranging from -1 to 1, signifying the presence or absence of a face, respectively.

5. Eliminating Overlapping Detections

Generally, the ANN classifier may produce multiple positive detections for a face, because the same face can be detected at multiple scales and at several nearby positions. This leads to multiple overlapped detections. It is clearly that eliminating these detections is one of the functions of the system. The face centers of the overlapped detections tend to cluster about a typical nearby region. Since faces rarely overlap in images; we propose to eliminate such detections using a familiar approach for pattern matching based on measures of distance between pattern vectors. The system computes a set of Euclidean distances $d_i(x)$ between each member of detections and our predefined sample face pattern (i.e. $d_1(x), d_2(x), \dots, d_n(x)$), if the i^{th} positive detection is identified as best match, then

$$d_i(x) < d_j(x) \quad j=1,2,\dots,n; j \neq i.$$

In other words, an unknown detection is said to be the best match if it yields the smallest Euclidean distance. The other overlapped detections are eliminated.

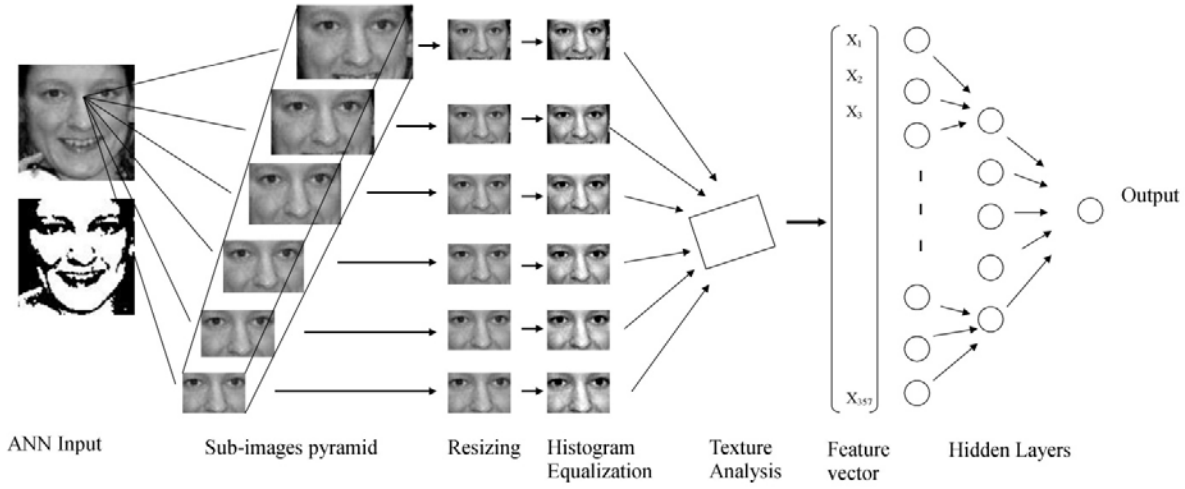


Figure 4. Neural network-based face detector Algorithm

6. Experimental Results

The system was tested on two sets of images. Test set A consists of 130 images containing single face. Test set B consists of 75 images containing multiple faces (115 faces). The faces are of different size, illumination, position and complex background. The images collected from three different databases: “The CVL Database” [18], “LFW database” [19] and our dataset contains 150 images are collected from the Web.

Three sets of experiments were performed to evaluate system performance: A conventional image-based feature vector, texture-based feature vector, and integrating of both image and texture analysis feature vector. The comparison among performance evaluations is shown in table 1. The table shows the effectiveness of the proposed face detector, which is based on integrating both image and texture features. Fig. 5 shows some face-detection results by our system.

When a “face” is detected in source image, the system draws an appropriately bounding box at the corresponding face. As there are single canonical face images with uniform background, the detection rate is very good. Considering the unconstrained nature of internet images containing many faces, the detection rate is slightly lower, but it is still good.

Compared to a well-known ANN-based face detection approach developed by Rowley et al. [13], our system is different with the following improvements:

First, the search space of Rowley’s approach is relatively high. The filter examines every location in the input image, whereas our system does not have this drawback. In this work, the search space is relatively low and restricted to portions of the image, which would speed up the algorithm as well as eliminating false detections.

Second, in this work, partial face pattern is used which minimizes face variations caused by beard, mustache and

mouth expressions; and consequently makes the training easier and therefore improves the detection accuracy. Third, texture descriptors improves system accuracy by eliminate many false detections, but at the expense of more computation. When detection methods are used within systems, it is important to consider both requirements, speed and accuracy. Accuracy may need to be sacrificed for speed or vice versa.

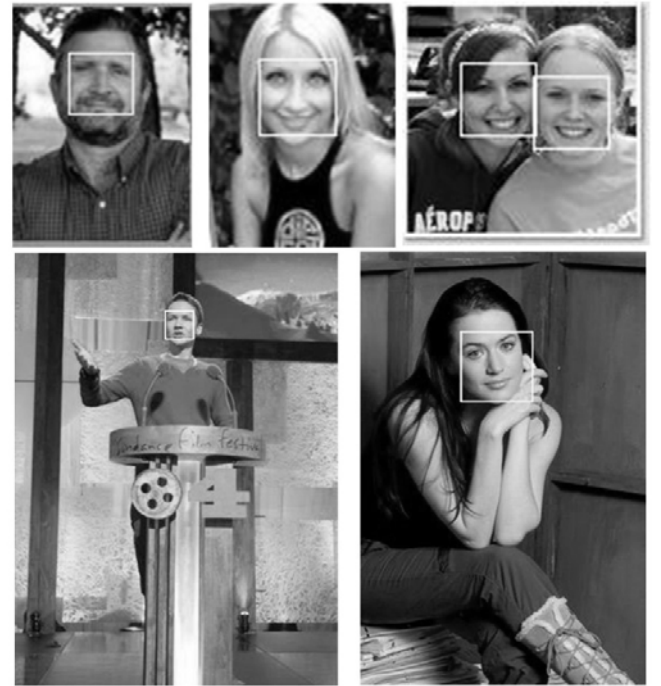


Figure 5. Some detection results of our system.

Table 1. Performance of ANN-based face detector using different feature vectors

| Method | Positive Detections | | False Detection | |
|--|---------------------|-------|-----------------|-------|
| | Set A | Set B | Set A | Set B |
| Image-based feature vector | 89.8% | 83.6% | 9.2% | 12.3% |
| Texture-based feature vector | 94.4% | 84.5% | 7.3% | 8.2% |
| Integrating image and texture feature vector | 95.3% | 89.5% | 4.4% | 6.1% |

7. Conclusion

In this paper, a novel system for human face detection in still gray images is presented. The system shows highly accurate results independent of scale, position, lighting condition and complex background.

We have used simple and efficient approach to segment the source image. A neural network-based face detector would then be applied to examine small windows of an image and decide whether each window contains a face. Texture descriptors such as mean, standard deviation, smoothness and X-Y-Relieves are measured and entered besides the image as input data to form solid feature vector.

ANN-based classifier is tested using three sets of experiments: A conventional image-based feature vector, texture-based feature vector and finally integration of both texture analysis and image feature vector.

Experimental results show that a combination of both pixel intensities and texture descriptors provide robust scheme for training and classification.

Image segmentation shows that 30% up to 52% of the search space is reduced following this method. Therefore, improvement in detection speed is achieved.

One limitation of the current system is that overlapping elimination fails in relatively few cases. The main limitation of the current system is that it only detects frontal faces with a $\pm 10^\circ$ rotation. In our future work, we can achieve higher rotation angle by rotating the cropped window to the proper angle and then passing it to the detection network. Handling rotated faces would impose additional computational cost.

References

- [1] Baskan S., Bulut M. M., and Atalay V., "Projection based method for segmentation of human face and its evaluation, Pattern Recognition Letters", vol. 23, 1623-1629 (2002).
- [2] Feraud R., O.J. Bernier, J.-E. Villet, and M. Collobert, "A Fast and Accurate Face Detector Based on Neural Networks," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, no. 1, pp. 42-53, Jan. 2001.
- [3] Gonzalez R. C., Woods R. E., "Digital Image Processing", Addison-Wesley, (2002).
- [4] Hiremath P. S. and A. Danti, "Detection Of Multiple Faces in an Image Using Skin Color Information And Lines-Of-Separability Face Model", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 20, No. 1 (2006) 39-6.
- [5] Hjelms E., "Face Detection: A Survey", Computer Vision and Image Understanding 83, 236-274 (2001).
- [6] Jin Z., Lou Z., Yang J., and Sun Q., "Face detection using template matching and skin-color information", Neurocomputing, Volume 70, Issues 4-6, 794-800 (2007).
- [7] Juang C-F., Shiu S-J., "Using self-organizing fuzzy network with support vector learning for face detection in color images", Neurocomputing , Volume 71 Issue 16-18 (2008).
- [8] Kouropoulos C., Pitas I., "Rule-Based Face Detection in Frontal Views", Proc. Int'l Conf. Acoustics, Speech and Signal Processing, vol. 4, 2537-2540 (1997).
- [9] Lin S.-H., S.-Y. Kung, and L.-J. Lin, "Face Recognition/Detection by Probabilistic Decision-Based Neural Network," IEEE Trans. Neural Networks, vol. 8, no. 1, pp. 114-132, 1997.
- [10] Moallem P., B. S. Mousavi, S. A. Monadjemi, "A novel fuzzy rule base system for pose independent faces detection", Applied Soft Computing 11 (2011) 1801-1810.
- [11] Nefian A. V. and M. H. H III, "Face Detection and Recognition Using Hidden Markov Models," Proc. IEEE Int'l Conf. Image Processing, vol. 1, pp. 141-145, 1998.
- [12] Osuna E., R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection", Computer Vision and Pattern Recognition, pp. 130-136, 1997.

- [13] Rowley H., S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 1, pp. 23-38, Jan. 1998.
- [14] Rowley H., S. Baluja, and T. Kanade, "Rotation Invariant Neural Network-Based Face Detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 38-44, 1998.
- [15] Russ J. C.: 'The Image Processing Handbook', 5th edition, Taylor & Frances Group, 2007.
- [16] Schneiderman H. and T. Kanade, A statistical method for 3D object detection applied to faces and cars, Proc. IEEE Conf. Computer Vision and Pattern Recognition (2000), pp. 746–751.
- [17] Shih F. y., S. Cheng and C. Chuang, "Extracting Faces And Facial Features From Color Images", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 22, No. 3 (2008) 515–534.
- [18] The CVL Database, Web address: <http://lrv.fri.uni-lj.si/facedb.html>.
- [19] The LFW Database Web Address: <http://vis-www.cs.umass.edu/lfw/>.
- [20] Turk M. and A. Pentland, "Eigenfaces for Recognition," J. Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.
- [21] Wang J., T. Tan, "A new face detection method based on shape information", Pattern Recognition Letters 21 (2000) 463-471.
- [22] Yang M.-H., D. J. Kriegman and N. Ahuja, Detecting faces in images: a survey, IEEE Trans. Patt. Anal. Mach. Intell. 24 (2002) 34–58.
- [23] Zainuddin R and Naji S. , "Multi-skin color clustering models for face detection", Second International Conference on Digital Image Processing, Singapore, 2010.
- [24] Zaqout I., R. Zainuddin and S. Baba, "Human Face Detection In Color Images" Advances in Complex Systems, Vol. 7, Nos. 3&4 (2004) 369–383.
- [25] Garcia C. and G. Tziritas, "Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis", IEEE Transactions on Multimedia, Vol. 1, NO. 3, September 1999